On the evaluation of the size of the answer of a relational query

Patrick Richard

INRIA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tél. 954 90 20

Rapports de Recherche

N° 51

ON THE EVALUATION OF THE SIZE OF THE ANSWER OF A RELATIONAL QUERY

Philippe RICHARD

Décembre 1980

ON THE EVALUATION OF THE SIZE OF THE ANSWER OF A RELATIONAL QUERY

PHILIPPE RICHARD
INRIA, ROCQUENCOURT (FRANCE)

Author's address : INRIA, B.P. 105, Domaine de Voluceau,
78153 Le Chesnay (France)

ON THE EVALUATION OF THE SIZE OF THE ANSWER OF A RELATIONAL QUERY

Philippe RICHARD

Résumé :

Nous présentons un modèle probabiliste pour évaluer la taille des relations construites par une requête exprimée dans l'algèbre relationnelle. Nous définissons les paramètres nécessaires à l'évaluation des relations dérivées ainsi que les hypothèses sur la structure probabiliste de la base.

La classe particulière de schémas de bases de données dans le cadre de laquelle nous évaluons la taille des réponses à une requête est caractérisée par des propriétés d'indépendance (entre deux relations distinctes ayant des domaines compatibles pour l'union ou entre deux n-uplets d'une même relation). Nous montrons que, connaissant l'espérance de la taille des projections des relations de la base de données, nous pouvons calculer la taille de toute requête exprimée dans l'algèbre relationnelle et nous donnons les résultats pour tous les opérateurs de ce langage (sélection, projection, union, intersection, θ-join).

Abstract :

We present a probabilistic model for evaluating the size of relations derived from given relations through relational algebra operators. We define tools to estimate the derived relations size and we state the assumptions underwhich we perform such an evaluation.

The particular class of data base schemata in which we evaluate the derived relation size is characterized by properties such as independence between two relations having union-compatible domains or independence between distinct tuples in a relation. We show that, knowing the expected size of the projection of each relation in the database, we can compute the size of each query expressed in relational algebra and we give the results for each operator of this language (selection, projection, union, intersection, θ-join).

INTRODUCTION

The relational approach provides data base management systems (DBMS) whose main feature is a great degree of data independence. In such systems, only the desired result must be specified by the user. Unlike the network or hierarchical models, no access path is specified. Thus an efficient query processing is particularly needed in relational DBMS. Optimizing query processing implies both the best choice of access paths and of the order in which relational operators must be processed. Thus, the optimizer needs a means to evaluate in advance, at each step of the query processing the size of the base and temporary relations. We illustrate by an example the importance of such an evaluation.

Let us consider the following relations :

EMP(NAME,#CHILDREN)
JOB(NAME,SALARY,DEPARTMENT)
LOC(DEPARTMENT,FLOOR)

Let us consider now the query : "Name,Department and Floor of each employee earning more than 5000, having more than 4 children and whose department is higher than the 1st floor" :

We can answer this query by two equivalent but different sequences of operations as follows :

- $[(EMP|\#CHILDREN > 4)[NAME = NAME](JOB|SALARY > 5000)]$
  $[DEPT = DEPT](LOC|FLOOR > 1)[NAME,DEPARTMENT,FLOOR]$

- $[(LOC|FLOOR > 1)[DEPT = DEPT](JOB|SALARY > 5000)]$
  $[NAME = NAME](EMP|CHILDREN > 4)[NAME,DEPARTMENT,FLOOR]$

In order to decide which sequence to use, we must know the size of each relation constructed at each step of the query processing. The first sequence is better than the second if the number of employees having more than four children is smaller than the number of Department located higher than the first floor. We need a means to evaluate in advance these numbers in order to decide which sequence of operations should be used.

The problem is the following :

Given a data base and a relational expression, how can we evaluate the size of the result without actually performing the operations.

We first construct a mathematical model. This model defines the general properties of the data base needed to evaluate the size of derived relations. We also define tools to perform such an evaluation. Section IV presents a particular class of data bases for which we can evaluate the expected size of derived relations in any query expressed in relational algebra. Section IV also contains an evaluation of derived relation size for all relational algebra operators.

Our results can be directly used in optimization algorithms involving the knowledge of derived relations size.

II. NOTATIONS

In this section, we briefly present our notation. A relation is represented by a relation descriptor $R(A_1:D_1,\ldots,A_n:D_n)$ where R_i is the relation name, the A_i's are the attributes names and the D_i's the associated domains. When there is no confusion, we shall omit the domains and simply write :

$$R(A_1,\ldots,A_n).$$

Sets of attributes will be denoted by letters such as U, V, W, X, Y.

A value of a relation descriptor $R(A_1,\ldots,A_n)$ is a finite set of n-tuples $(x_1,\ldots,x_n)$ where each $x_i \in D_i$. We denote it by subscripting the relation name, i.e : $R_t(A_1,\ldots,A_n)$. The projection of R(U) on $X \subseteq U$ is denoted by R(U)[X], the selection of R(U) on the expression E, by R(U)|E and finally the θ-join of R(U) and R'(V) on $X \subseteq U$ and $Y \subseteq V$, by $R(U)[X \theta Y]R'(V)$. A data base consists of one or several relations which take at each time specific values. So a data base is caracterized by the set $R = \{R_1,\ldots,R_p\}$ of relation descriptors. $\mathbb{R}$ is the data base descriptor. The set of values that the relations can take is the data base schema denoted by S.

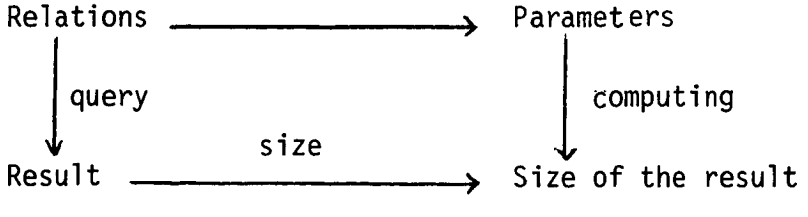
Data base values are denoted by $\mathbb{R}_t$. Usually the data base schema is caracterized by a set of integrity constraints. One type of integrity constraints are functional dependencies, defined as follows : Let $R_t(U)$ be a relation value and (X,Y) two subsets of U. We say that Y functionally depends on X iff :

$$\forall (x,x') \in (R_t(U))^2,\ x(X) = x'(X) \Rightarrow x(Y) = x'(Y)$$

We denote : $X \xrightarrow[R_t]{} Y$ or $X \rightarrow Y$ when there is no confusion.

III. The Model

As we have seen in the introduction, our problem is to evaluate the size of the result of a query in order to optimize its processing. We illustrate this problem as follows :



We wish to evaluate the size of the result in advance. Thus, it is necessary to have parameters which describe the base relations. These parameters can be, for example, the expected sizes of the base relations and the selectivity of their attributes. These parameters will give us a probabilistic evaluation of the size of the result of a query. Thus we need a probabilistic description of data base schemata. The most important is the choice of elementary events and we choose for these, occurences of data base values. Let S be a data base schema. We call probabilistic schema the pair (S,p) where p is a probability measure such that :

$$— R_t \longrightarrow p[R = R_t]$$

From such a probabilistic schema, we can deduce all the other probabilities. Let us consider the following example : Let S be a data base schema consisting of the values of the relation R(EMPLOYEE, DEPARTMENT).
Let E be the following event :

"The number of employees of the shoes department is equal to 10".

The probability of E is :

$$p(E) = \sum_{R_t \text{ such that } E \text{ is true in } R_t} p(R = R_t)$$

That means that the probability of E is the sum of the probabilities of the relation values in which the number of employees of the shoes department is equal to 10. In general, we can not define the probability measure p. We just introduce it as a theoretical tool and we shall see that all we need to know are some properties of this measure (independence, distribution, etc...). A parameter system can be represented as a mapping that from the data base values computes the actual values of the parameters.

The parameters are provided by the DBMS and periodically updated. From these parameters, the optimizer can compute the size of base relations and of derived relations. We have to find a "good" parameter system, which allows us to estimate derived relation sizes for each type of query expressed in relational algebra. This will be formalized later in the notion of validity of a parameter system. But, before, let us consider the following example.

Let S be the schema consisting of the values of

R(EMPLOYEE, DEPARTMENT, MANAGER)

These values must satisfy the following constraints :

EMPLOYEE → DEPARTMENT
DEPARTMENT → MANAGER
MANAGER → DEPARTMENT

Let us consider a relation value R_t(EMPLOYEE, DEPARTMENT, MANAGER) of our schema. Let us consider the parameter system that consists of the expected size of particular projections of R.

Let E_{EMP} be the expected size of the projection of R_t on "EMPLOYEE". Define similarly E_{DEPT} and E_{MGR} for "DEPARTMENT" and "MANAGER". The mapping $P : S \rightarrow N^3$ such that $P(S) = \{E_{Emp}, E_{Dept}, E_{Mgr}\}$ is a parameter system.

Given P, we can compute the expected values of any projection of the relation. Indeed, due to the functional dependencies, we have :

$$|R(\text{EMPLOYEE, DEPARTMENT, MANAGER})| = |R(\text{EMPLOYEE})| =$$

$$|R(\text{EMPLOYEE, DEPARTMENT})| = |R(\text{EMPLOYEE, MANAGER})|$$

and $$|R(\text{DEPARTMENT, MANAGER})| = |R(\text{MANAGER})|$$

Therefore P is appropriate for the projection operator in our example.

During the query processing, derived relations will be constructed from base relations. In order to evaluate the size of the result

of the query, we need parameters on data base relations and on intermediate derived relations too. A parameter system defines characteristics on data base relations, but we need the same characteristics on derived relations at each step of a query processing. These characteristics must be deduced from the initial ones.

In general, it is not possible, and we shall study particular probabilistic schemata and parameter system that propagates through relational operators. We shall call, this property of parameter systems, validity.

Let us consider the schema and the parameter system defined in the example above :

$$P(S) = \{E_{EMP}, E_{DEPT}, E_{MGR}\}$$

Due to the integrity constraints in S, P obviously propagates through projection operator. Indeed, we can compute the size of any query containing only projections.

Let us now consider, the following query :

$$R' = R(EMPLOYEE, DEPARTEMENT, MANAGER) \mid DEPARTMENT = \text{"shoes"}$$

$$E|R'| = \begin{cases} 0, \text{ if "shoes" does not belong to R} \\ \dfrac{E|R(EMPLOYEE,\ DEPARTMENT,\ MANAGER)|}{E|R(DEPARTMENT)|} \text{ otherwise} \end{cases}$$

Thus, our parameter system does not propagate through selection operator. We need a new parameter which does not belong to P and can not be deduced from P :

$$Prob(\text{"shoe"} \in R(DEPARTMENT))$$

In order to solve this problem, we need parameter systems which either contain these probabilities or which allow us to evaluate these probabilities according to particular properties of data base schemata.

The first solution is more general than the second but the appropriate parameter systems are rather complex and results are mainly theoretical as shown in [RICH 80]. In this paper, we shall be concerned only with the second solution which gives nice results which can be used in optimization algorithms.

We can now formalize the notion of validity of a parameter system [RICH 80].

We denote by $S + \varphi(S)$ the schema containing the relation values of S and those obtained by applying the operator φ to one or two relations values of S. (φ is projection, selection, θ-join, union, intersection).

Definition III.2

Let φ be a relational operator.
Let C be a class of probabilistic schemata closed under φ.
Let $\mathcal{P}$ be a parameter system.
$\mathcal{P}$ is valid in C for φ if and only if :

$\forall\ (S, p) \in C,$

1) $\exists f$ such that : $\forall\ IR_t \in S.$

$$E|\varphi(IR_t)| = f(\mathcal{P}(IR_t))$$

2) $\exists g$ such that : $\forall\ IR_t \in S$

$$\mathcal{P}(IR_t + \varphi(IR_t)) = g(\mathcal{P}(IR_t)). \quad \square$$

This definition mainly states that the following diagrams commutes.

$$\begin{array}{ccc} S & \xrightarrow{\mathcal{P}} & \mathcal{P}(S) \\ \downarrow{\scriptstyle \varphi} & & \downarrow{\scriptstyle f} \\ \varphi(S) & \xrightarrow{E||} & f(\mathcal{P}(S)) \end{array}$$

$$\begin{array}{ccc} S & \xrightarrow{\mathcal{P}} & \mathcal{P}(S) \\ \downarrow{\scriptstyle id+\varphi} & & \downarrow{\scriptstyle g} \\ S+\varphi(S) & \longrightarrow & g(\mathcal{P}(S)) \end{array}$$

Roughly speaking, (1) expresses that the parameter system allows us to compute the expected size of derived relation constructed by applying any relational operator to S. (2) means that $P(S+\varphi(S))$ can be deduced from $P(S)$. i.e. that we can compute the parameters of the new "augmented" data base.

Indeed, evaluating a complex query implies that we are able to evaluate for each operator of this query the size of its result and mainly the parameters on the derived relation which is constructed by this operation.

For example, if our query contains first a selection and after a projection, we must have the same parameters on the relation constructed by the selection as on the base relation, otherwise, we may not compute the size of the projection. If our parameters are the expected size of any projection of the base relation, we must compute the expected size of any projection of the derived relation. Furthermore, we have assumed that our schema satisfies a set of properties (since it belongs to a class), thus the new schema which contains the old one and the new relation values (say the ones obtained by selection) must satisfy the same properties, otherwise we can not perform evaluation on the rest of the query. That is, our class must be closed.

The schema itself is a valid parameter system, where the parameters are the relation values, for all relational operators.
We notice that, for general schemata, 1 and 2 are not true. We have seen in the example above, that the system was not valid for the selection in the most general class (which contains all schemata), but that it was for the projection.

IV. Application and results

In this section, we define a particular class of probabilistic schemata and a parameter system. Then we show that this system is valid for the class and for each relational algebra operator. We give results of the evaluation of the size of derived relations constructed by projection, selection, union, intersection and join.

A/ Universes and conditions

We define the conditions which caracterize our class of schemata. Our goal is to find parameter systems providing results on the query evaluation which can be used by optimization algorithms. General probabilistic schemata can not provide us such a feasibility. Furthermore, in the real world, particular properties of the data base allow us to simplify, in a realistic way, the model which we study. It was seen before, that evaluating probabilities such as P("shoes" ϵ R(Department)) is necessary. Generally the notion of domain does not allow us to compute this probability.
So, we have to define an other notion.

Definition IV.1

Let R(U) be a relation
The universes of R(U) (denoted by $\bar{R}(U)$) are a finite collection of disjoints sets which satisfy the two following axioms :

A1 : $|\bar{R}(U)| < \infty$

A2 : $\forall R_t(U) : R_t(U) \subseteq \cup\bar{R}(U)$. □

Generally we define the universes of each simple attribute of U. For a subset X of U, universes of R(U)[X] are obtained by performing the cartesian product of the universes of R(U)[A] where A ϵ X. We illustrate by an example this definition and we show the need of a collection of universes rather than a unique set.

Let us consider the following relation :

EMP(NAME,DEPARTMENT,SALARY)

The universe of NAME is the set of all the names of the general store employees denoted by $\overline{EMP}$(NAME). The universe of DEPARTMENT is {toys, shoes,clothes}. The universe of SALARY is the set of natural numbers greater than 1500 and smaller than 8000 ($\overline{EMP}$(SALARY)). The universe of "NAME,DEPARTMENT,SALARY" is the cartesian product of the universe of "NAME", "DEPARTMENT" and "SALARY".

Let us consider the following operation :

$$R = EMP\,|\,DEPARTMENT = \text{"shoes"} \vee SALARY = \text{"2500"}$$

We can write :

$$R = (EMP\,|\,DEPARTMENT = \text{"shoes"}) \cup (EMP\,|\,SALARY = \text{"2500"})$$

The universe of $(EMP\,|\,DEPARTMENT = \text{"shoes"})$ is :

$$\overline{EMP}(NAME) \times \{shoes\} \times \overline{EMP}(SALARY)$$

The universe of $(EMP\,|\,SALARY = \text{"2500"})$ is :

$$\overline{EMP}(NAME) \times \overline{EMP}(DEPARTMENT) \times \{2500\}$$

$(EMP\,|\,DEPARTMENT = \text{"shoes"})$ and $(EMP\,|\,SALARY = \text{"2500"})$ are not independent in the sense that some tuples may exist in one of them and not in the other.

So, the universes of R must be defined as follows :

1) $\overline{EMP}(NAME) \times \{shoes\} \times \{2500\}$ contains all tuples which can exist in both relations.

2) $\overline{EMP}(NAME) \times \{shoes\} \times (\overline{EMP}(SALARY) - \{2500\})$ contains all tuples which can exist only in $(EMP\,|\,DEPARTMENT = \text{"shoes"})$

3) $\overline{EMP}(NAME) \times (\overline{EMP}\,|\,DEPARTMENT) - \{shoes\}) \times \{2500\}$ contains all tuples which can appear only in $(EMP\,|\,SALARY = \text{"2500"})$

These three sets are the collection of universes of R. Therefore, during the evaluation of the size of derived relations, it may be necessary to create (for each new derived relation implied at each step of the query processing) a collection of disjoint universes to express new interrelational dependencies. So we do not evaluate the size of the derived relation but the size of each intersection of this relation with each of its universes. The whole size of the derived relation will obtained by summing all these results since universes are disjoint sets.

We justify the notion of universes for a relation as follows:

- generally, the set of values which can arise in the data base is known : if we consider the example above, we can check the set of possible salaries in the store and the different departments can be listed.

- universes do not change quickly :
  The store does not create or suppress department too frequently during the data base life.

Now we may define the three conditions which caracterize the class of probabilistic schemata that we shall study later.

Definition IV.2

Let (S,p) be a probabilistic schema.
We say that (S,p) satisfies the equi-probability condition if :

$\forall\, R_t(U) \in S,\ \forall\, X \subseteq U,\ \forall\, (x,x') \in \bar{R}(X)^2$:

$$\text{Prob}(x \in R_t(X)) = \text{Prob}(x' \in R_t(X))$$

(where $\bar{R}(X)$ is the union of all the universes of $R(X)$). □

If we consider the example above, we may suppose that the probabilities that "shoes", "toys", or "clothes" belong to the relation Emp(Department) are the same.

Definition IV.3

Let (S,p) be a probabilistic schema :
Let R(U) and R'(U') be two relations in this schema.
Let $X \subseteq U$ and $Y \subseteq U'$ be two sets whose universes are union-compatible.
Let $\bar{R}(X)$ and $\bar{R}'(Y)$ be two non disjoint universes of R(X) and R'(Y).
R(X) and R'(Y) are independent for $\bar{R}(X)$ and $\bar{R}'(Y)$ iff :

$\forall\, t,\ \forall\, (x,y) \in (\bar{R}(X) \cap \bar{R}'(Y))^2$: $(x \in R_t(X))$ is independent of $(y \in R'_t(Y))$. □

This condition depends on the universes $\bar{R}(X)$ and $\bar{R}'(Y)$. Two relations may be independent for particular pairs $(\bar{R}(X), \bar{R}'(Y))$ and not for others.

In order to simply formalize the third condition, we state the following definition :

Definition IV.4

Let S be a schema.
Let u and u' be two distinct tuples.
We say that u and u' are non compatible iff :

$\forall R_t(U) \in S : u \in R_t(U) \Rightarrow u' \notin R_t(U)$
and $u' \in R_t(u) \Rightarrow u \notin R_t(U)$. □

Let us consider the relation Emp(NAME,DEPARTMENT,MANAGER). The tuples "Toto,Toys,Jojo" and "Toto,Shoes,Lili" can not co-exist in the same relation value, because an employee works in only one department. Functional dependencies tell us which tuples are non compatible as the following lemma shows

Lemma IV.1

Let S be a schema and $\mathcal{F}$ its set of FD's.
Let R(U) be a relation and u and u' be two distinct tuples of R.
u and u' are non compatible iff :
$\exists$ A, B $\subseteq$ U, A $\neq$ B such that $(A \rightarrow B) \in \mathcal{F}$ and u(A) = u'(A) and u(B) $\neq$ u'(B). □

Proof :

Suppose there exists a non trivial dependency $(A \rightarrow B) \in \mathcal{F}$ such that $u(A) = u'(A)$ and $u(B) \neq u'(B)$.
Then any relation value which contains both u and u' breaks the dependency and can not belong to the schema.

Conversely, suppose that, $\forall A,B \; / \; A \rightarrow B \in \mathcal{F}$, $u(A) = u'(A) \Rightarrow u(B) = u'(B)$. Then, from the definition of S, there exist relation values to which both u and u' belongs.

We now define our class of probabilistic schemata.

Definition IV.5

We call CE the class of probabilistic schemata (S,p) which satisfies the three following statements :

1) (S,p) satisfy the equi-probability condition.

2) inter-relational independence :

$\forall (R_t(U), R'_t(U')) \in S^2$, $\forall X \subseteq U$ and $Y \subseteq U'$ such that X and Y are associated to union-compatible universes
$\forall \bar{R}(X)$ and $\bar{R}'(Y)$ two non disjoint universes of R(X) and R'(Y), one of the following is true :

i) $R_t(X)$ and $R'_t(Y)$ are independent for $\bar{R}(X)$ and $\bar{R}'(Y)$

ii) $R_t(X) \cap \bar{R}'(Y) \cap \bar{R}(X) \subseteq R'_t(Y) \cap \bar{R}(X) \cap \bar{R}'(Y)$

iii) $R_t(X) \cap \bar{R}'(Y) \cap \bar{R}(X) \subseteq R'_t(Y) \cap \bar{R}(X) \cap \bar{R}'(Y)$

3) intra-relational independence :

$\forall R_t(U) \in S$, $\forall X \subseteq U$
$\forall (x,x') \in \cup\bar{R}(X)$,
x and x' compatible $\Rightarrow (x \in R_t(X))$ and $(x' \in R_t(X))$ are independent.

We now define a parameter system which we shall use to estimate the expected size of derived relations.

Let (S,p) be a probabilistic schema. We shall denote by P, the system that associates to (S,p) the expected size of all projections of each relation in S.

B/ Results

We show that this system allows us to evaluate the expected size of derived relations obtained by applying any operator of Codd's relational algebra on the data base. Furthermore, since P is valid in CE for Codd's algebra, we can compute the size of derived relations constructed by any query expressed in this algebra.

In order to prove this statement, we first show that CE is closed for each operators of Codd's algebra. We recall that these operators are projection, selection, union, intersection and join [CODD 71].

Theorem IV.1

Let CE be the class defined in IV.5.
CE is closed for Codd's algebra. □

Proof :

For each operator φ, we have to show that for any probabilistic schema (S,p) in CE, S+φ(S) is also in CE. That is, the new schema (which contains the derived relation constructed by φ) satisfies the three properties of CE.

The details of the proof are given in [RICH 80].

We now prove that P is valid in CE for the relational algebra.

Theorem IV.2

Let CE be the class defined in IV.5.
Let P be the system defined in IV.6.
P is valid in CE for the relational algebra. □

Proof :

In order to prove this theorem, we have to show two statements :

1) For each operator φ, we must show that we can check the expected size of the relation constructed by φ from the parameter system.

2) For each operator φ, we must construct the parameter system associated to $S+\varphi(S)$ from the parameters associated to S and from the definition of φ.

For our system, (1) is a particular case of (2). But there exist systems for which (1) and (2) are quite different. See [RICH 80] for examples of such systems and for details of the proof .

In the following tables, we give for each operator, the definition of this operator, the parameters needed and the expected size of the result. In case of binary operator, we give the expected size of the projection of the derived relation on its universes (due to the inter-relational dependencies). These results are directly available for optimization algorithms which need an evaluation of derived relations size.

Operators	Parameters	Expected size of the result
projection $R(U)[X]$	$E\|R(X)\|$	$E\|R(X)\|$
selection $R(U)\|X = x$	$E\|R(U)\|$	$\frac{E\|R(U)\|}{\|\cup\bar{R}(X)\|}$
selection $R(U)\|X > x$	$E\|R(U)\|$	$\frac{n\ E\|R(U)\|}{\|\cup\bar{R}(X)\|}$ where $n = \|\cup\bar{R}(X)\|X > x\|$
selection $R(U)\|X \neq x$	$E\|R(U)\|$	$\frac{E\|R(U)\|}{\|\cup\bar{R}(X)\|} \times (\|\cup\bar{R}(X)\| - 1)$
selection $\|R(U)\|X = Y$	$E\|R(U)\|$	$\frac{E\|R(U)\|}{\|\cup\bar{R}(XY)\|} \times \|(\cup\bar{R}(X)) \cap (\cup\bar{R}(Y))\|$

Operators	Parameters	Expected size of the result
selection $R(U) \mid X > Y$	$E\lvert R(U)\rvert$	$\frac{E\lvert R(U)\rvert}{\lvert \cup \bar{R}(U)\rvert} \times \sum_{y \in \cup \bar{R}(Y)} n_y$ where $n_y = \lvert \{x \in \cup \bar{R}(X) / x > y\}\rvert$
intersection $R(U) =$ $R'(U) \cap R''(U)$	$E\lvert R'(U)\rvert$ $E\lvert R''(U)\rvert$	1) inter-relational independence of $R'(U)$ and $R''(U)$ for $\bar{R}'(U)$ and $\bar{R}''(U)$. $E\lvert R(U) \cap \lvert \bar{R}'(U) \cap \bar{R}''(U)\rvert = \lvert \bar{R}'(U) \cap \bar{R}''(U)\rvert \times \frac{E\lvert R'(U)\rvert \times E\lvert R''(U)\rvert}{\lvert \cup \bar{R}'(U)\rvert \times \lvert \cup \bar{R}''(U)\rvert}$ 2) $R'(U) \cap \bar{R}''(U) \cap \bar{R}'(U) \subseteq R''(U) \cap \bar{R}''(U) \cap \bar{R}'(U)$ $E\lvert R(U) \cap \bar{R}'(U) \cap \bar{R}''(U)\rvert = \lvert \bar{R}'(U) \cap \bar{R}''(U)\rvert \times \frac{E\lvert R'(U)\rvert}{\lvert \cup \bar{R}'(U)\rvert}$
Union $R(U) =$ $R'(U) \cup R''(U)$	$E\lvert R'(U)\rvert$ $E\lvert R''(U)\rvert$	$E\lvert R(U) \cap \bar{R}'(U) \cap \bar{R}''(U)\rvert = E\lvert R'(U) \cap \bar{R}'(U)\rvert + E\lvert R''(U) \cap \bar{R}''(U)\rvert - E\lvert R'(U) \cap R''(U) \cap \bar{R}'(U) \cap \bar{R}''(U)\rvert$

Operators	Parameters	Expected size of the result
equi-join $R(XYZ) = R'(XY)[X=\bar{X}]R''(XZ)$	$E\|R'(X)\|$, $E\|R''(X)\|$ $E\|R(XY)\|$, $E\|R''(XZ)\|$	1) Intra-relational independence in $R'(X)$ and $R''(X)$ and inter-relational independence of $R'(X)$ and $R''(X)$ for $\bar{R}'(X)$ and $\bar{R}''(X)$ $E\|R(XYZ) \cap [(R'(X) \cap R''(X)) \times \cup\bar{R}'(Y) \times \cup\bar{R}''(Z)]\|$ $= \|R'(X) \cap R''(X)\| \times \frac{E\|R'(XY)\| \times E\|R''(XZ)\|}{\|\cup\bar{R}'(X)\| \times \|\cup\bar{R}''(X)\|}$ $* \; E\|[R(XYZ) \cap [(R'(X) \cap R''(X)) \times \cup\bar{R}'(Y) \times \cup\bar{R}''(Z)]][X'YZ]\|$ $= \frac{\|\cup\bar{R}(X'YZ)\|^2}{\|\bar{R}(X'YZ)\|} \times [1 - [1 - \frac{E\|R'(XY)\| \times E\|R''(XZ)\| \times R(X'YZ)\|}{\|\cup\bar{R}(X'YZ)\|^2 \times \|\bar{R}'(X)\| \times \|\bar{R}''(X)\|}]^{\|\bar{R}'(X) \cap \bar{R}''(X)\|}]$ where $\bar{R}(X'YZ) = [(\bar{R}'(X) \cap \bar{R}''(X)) \times (\cup\bar{R}'(Y)) \times (\cup\bar{R}''(Z))][X'YZ]$ where $X' \subsetneq X$ and $X' \neq X$

Operators	Parameters	Expected size of the result
		2) Intra-relational independence in $R'(X)$ and $R''(Y)$ and $R'(X) \cap \bar{R}'(X) \cap \bar{R}''(X) \subseteq \bar{R}'(X) \cap \bar{R}''(X) \cap R''(X)$ $\bar{R}(XYZ) = (\bar{R}'(X) \cap \bar{R}''(X)) \times \cup\bar{R}'(Y) \times \cup\bar{R}''(Z)$ $E\lvert R(XYZ) \cap \bar{R}(XYZ)\rvert = \lvert\bar{R}'(X) \cap \bar{R}''(X)\rvert \times \frac{E\lvert R'(XY)\rvert \times E\lvert R''(XZ)\rvert}{\lvert\cup\bar{R}'(X)\rvert \times E\lvert R''(X)\rvert}$ $* E\lvert[R(XYZ) \cap \bar{R}(XYZ)][X'YZ]\rvert = \frac{\lvert\cup\bar{R}(X'YZ)\rvert^2}{\lvert\bar{R}(X'YZ)\rvert} \times$ $[1 - [1 - \frac{E\lvert R'(XY)\rvert \times E\lvert R''(XZ)\rvert \times \lvert\bar{R}(X'YZ)\rvert}{\lvert\cup\bar{R}(X'YZ)\rvert^2 \times \lvert\cup\bar{R}(X)\rvert \times E\lvert R'(X)\rvert}]^{\lvert\bar{R}'(X) \cap \bar{R}''(X)\rvert}]$ 3) $\exists(U,V)$ a partition of X such that $U \rightarrow V$ in $R'(X)$ or $R''(X)$ and $R'(X)$ and $R''(X)$ are independent for $\bar{R}'(X)$ and $\bar{R}''(X)$ $\bar{R}(U) = \bar{R}'(U) \cap \bar{R}''(U)$ and $\cup\bar{R}(V) = ((\cup\bar{R}'(V)) \cap (\cup\bar{R}''(V)))$ $E\lvert R(XYZ) \cap \bar{R}(XYZ)\rvert = \lvert\bar{R}(U)\rvert \times \lvert\cup\bar{R}(V)\rvert \times \frac{E\lvert R'(XY)\rvert \times E\lvert R''(XZ)\rvert}{\lvert\cup\bar{R}'(X)\rvert \times \lvert\cup\bar{R}''(X)\rvert}$

Operators	Parameters	Expected size of the result
		* $E\lvert R(XYZ) \cap \bar{R}(XYZ)\ [X'YZ]\rvert = \dfrac{\lvert\bar{R}(X'YZ)\rvert^2}{\lvert\cup\bar{R}(X'YZ)\rvert}$ $\left[1 - \left[1 - \dfrac{\lvert\cup\bar{R}(V)\rvert \times E\lvert R'(XY)\rvert \times E\lvert R'(XZ)\rvert \times \lvert\bar{R}(X'YZ)\rvert}{\lvert\cup\bar{R}'(X)\rvert \times \lvert\cup\bar{R}''(X)\rvert \times \lvert\bar{R}(X'YZ)\rvert^2}\right]^{\lvert R''(U)\rvert}\right]$ * These results are an example of those needed to show that our system is valid.

CONCLUSION

There have been already some evaluation of derived relation size but it was never clear under which assumptions the method worked.

One of the main contributions of this paper is the clear statement of the conditions underwhich the parameter system is valid. These conditions (especially equi-probability) appear in fact to be fairly strong and it is shown [RICH 80] that they are necessary.

So we have defined a class of particular data bases and a system of parameter on this class. We have shown that this system allows us to evaluate the expected size of derived relations resulting from any query expressed in algebra and we have given the expected size of derived relations for each operator of this algebra.

RÉFÉRENCES

[CHIU 80] Chiu D.M., Ho Y.C., "A Methodology for Interpreting Tree Queries into Optimal Semi-join Expression". Proc. of ACM-SIGMOD 1980, Int. Conf. on Management of Data, May 14-16.

[CLAU 80] Clausen S.E., "Optimizing the evaluation of calculus expressions in a relational database system", Inform. Systems, Vol. 5, pp. 41-54, 1980.

[CODD 70] Codd E.F., "A relational model for large shared data banks", Commun. ACM 13, 6 (June 1970), 377-387.

[CODD 71] Codd E.F., "Relational completeness of data base sub-languages", in Data Base Systems, Courant Computer Science Symposia, Vol. 6, Prentice-Hall, Englewood Cliffs, N.J., May 1971.

[DEMO 80] Demolombe E., "Estimation of the number of tuples satisfying a query expressed in predicate calculus language", O.N.E.R.A. C.E.R.T. VLDB Montréal Canada, Oct. 1-3, 1980.

[EPST 79] Epstein R., Stonebraker M., Wong E., "Distributed query processing in a relational data base system", ACM-SIGMOD 79.

[RICH 80] Richard P., "Evaluation de la taille d'une requête" Thèse 3ème cycle en cours, Paris-Sud (France).